UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/749,740	12/30/2003	Nikolai G. Nikolov	6570P040	9016

45062   7590   04/15/2009
SAP/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER
TECKLU, ISAAC TUKU

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
04/15/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _27 January 2009_.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1,3-10,13-15,22,24,26,28,30,31,43-45 and 47_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45 and 47_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _01/27/09_.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   This action is responsive to the amendment filed on 01/27/2009.

2.   Claims 1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45 and 47 have been examined.

Continued Examination Under 37 CFR 1.114

3.   A request for continued examination under 37 CFR 1.114, including the fee set forth in

37 CFR 1.17(e), was filed in this application after final rejection. Since this application is

eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e)

has been timely paid, the finality of the previous Office action has been withdrawn pursuant to

37 CFR 1.114. Applicant's submission filed on 01/27/2009 has been entered.

Response to Arguments

4.   Applicant's arguments with respect to claims 1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45

and 47   have been considered but are moot in view of the new ground(s) of rejection. See

Boykin et al. (US 2004/0123279) in view of Bley et al. (US 7,496,896 B2, new art made of

record) and Boykin et al. (US 2004/0123279) in view of Berry et al. (US 6,742,178 B1, new art

made of record) below.

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person

having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6.   Claims 1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45 and 47 are rejected under 35 U.S.C.

103(a) as being unpatentable over Boykin et al. (US 2004/0123279 A1) in view of Bley et al.

(US 7,496,896 B2).

    Per claim 1 (Currently Amended), Boykin discloses a <u>method for modifying a software</u>

<u>application</u>, comprising:

    <u>modifying</u> a classfile after said classfile has been compiled from source code, <u>version of</u>

<u>software application </u>said classfile describing properties of a class within an object oriented

environment (e.g. FIG. 4B, 412, 414 and related text);

    modifying a method information structure <u>for each method associated with the software</u>

<u>application</u> by adding byte code instructions to said method information <u>structure</u> (e.g. FIG. 11A-

11B and paragraph [0007] "… injector inserts hooks in the loaded class…" and paragraph [0055]

– [0055]), to cause a plug-in handler method <u>associated with a plug-in handler</u> to execute an

output function for <u>each</u> method<u>, the plug-in handler to record method information associated</u>

<u>with methods </u>(e.g. Table 9, line 12, receiving said identifier "handle"; line 20, invoke

'hooks.pos(handle, new Integer(returnValue))'; Table 6, details of "hooks.post", "probe.post",

which calculates/reports transfer rate about the "read" method); and,

    ~~adding a method information structure that includes byte code instructions for registering~~

~~the identities of said class and said method with a dispatch unit that is responsible for dispatching~~

~~an invocation to said plug-in module during runtime execution of said modified byte code, said~~

~~invocation directed to said dispatch unit from said added byte code instructions~~

compiling results of the modifying of the class file, the results including method

information, the method information including dependency hierarchical tree indicting

dependency order of the methods, and a time hierarchical tree indicting chronological order of

the methods (paragraph [0079] "… add the special instrumentation code and then recompile the

modified class…" and e.g. FIG. 4B and related text); and

filtering the method information via a filtering module, according to user preferences and

the dependency and time hierarchical tree (paragraph [0054] "… probed classes can be specified

individually or through filters…" and "… methods can be specified individually or through

filters such as all 'public' methods or all methods with parameter list and e.g. FIG. 11A and

related text),

Boykin does not explicitly disclose inserting functions calls at entry points and exit points

of each method associated with the software application via a byte code modifier. Nevertheless,

as evidenced by the teaching of Bley, it is known to insert function calls at entry and exit points

of each method via a byte code modifier (see at least e.g. Fig. 5, steps 302-314). Thus, it is

respectfully submitted that it would have been obvious to one skilled in the art at the time the

invention was made to insert function calls at entry and exit points of each method to add

functionality such as stopping a timer, or record information and perform action at the entry and

exit portion of the byte code as once suggested by Bley (see at least col.9:30-50).

Per claim 3 (Currently Amended), Boykin discloses the method of claim 2 <u>further</u>

<u>comprising:</u>

adding a field information structure <u>to the method,</u> said field information structure

describing a field that is to store a numeric identifier of said class (paragraph [0071] "... inserts

code at ..." and e.g. FIG. 7, step 708).

Per claim 4 (Currently Amended), Boykin discloses the method of claim 3 wherein said

numeric identifier is provided to said class by a method of which <u>a</u> dispatch unit is comprised

(see e.g. FIG. 4, registry e.g. by providing an identifier and Table 12).

Per claim 5 (Currently Amended), Boykin discloses the method of claim 1 wherein a

portion of said byte code instructions that are added to said method are for causing said plug-in

module's handler method to provide <u>an</u> output function treatment in response to an entry point of

said method being reached (e.g. Table 9, line 12, receiving said identifier "handle"; line 20,

invoke "hooks.pos(handle, new Integer(returnValue)).

Per claim 6 (Currently Amended), Boykin discloses the ~~classfile modification~~ method of

claim 5 wherein said output function treatment is a function selected from the group consisting

of:

1) recording a time of entry for said method (Table 6 getEntries(); for (int i = 0)... i++);

2) recording an input parameter value for said method (Table 6 getEntries(); for (int i =

0)... i++); and,

3) incrementing a counter for said method (Table 6 getEntries(); for (int i = 0)... i++).

Per claim 7 (Currently Amended), Boykin discloses the classfile modification method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in ~~module's~~ handler method to provide said output function treatment in response to an exit point of said method being inevitably reached (e.g. Table 9, line 12, receiving said identifier "handle"; line 20, invoke "hooks.pos (handle, new Integer(returnValue)).

Per claim 8 (Currently Amended), Boykin discloses the method of claim 7 wherein said output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method (Table 6 getEntries(); for (int i = 0)... i++);

2) recording an input parameter value for said method (Table 6 getEntries(); for (int i = 0)... i++); and,

3) incrementing a counter for said method (Table 6 getEntries(); for (int i = 0)... i++).

Per claim 9 (Currently Amended), Boykin discloses the method of claim 7 wherein portions of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to any exit point of said method being inevitably reached (e.g. Table 9, line 12, receiving said identifier "handle"; line 20, invoke "hooks.pos(handle, new Integer(returnValue)).

Per claim 10 (Currently Amended), Boykin discloses the method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an error arising

during execution of said method (e.g. Table 9, line 12, receiving said identifier "handle"; line 20, invoke "hooks.pos(handle, new Integer(returnValue)).

Per claim 13 (Currently Amended), Boykin discloses the method of claim 12 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke special instruction (paragraph [0079] "… add the special instrumentation code and then recompile the modified class…" and e.g. FIG. 4B and related text).

Per claim 14 (Currently Amended), Boykin discloses the method of claim 1 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke virtual instruction (see Table 3 and paragraph [0079] "… add the special instrumentation code and then recompile the modified class…" and e.g. FIG. 4B and related text).

Per claim 15 (Currently Amended), Boykin discloses the classfile modification method of claim 1 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke special instruction (see Table 3 and paragraph [0079] "… add the special instrumentation code and then recompile the modified class…" and e.g. FIG. 4B and related text).

Per claim 22 (Currently Amended), this is the machine readable storage medium version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 24 (Currently Amended), this is the machine readable medium version of the claimed method discussed above (Claim 3), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 26 (Currently Amended), this is the machine readable storage medium version of the claimed method discussed above (Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 28 (Currently Amended), this is the machine readable storage medium version of the claimed method discussed above (Claim 7), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 29 (Currently Amended), this is the machine readable storage medium version of the claimed method discussed above (Claim 8), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 30 (Currently Amended), this is the machine readable storage medium version of the claimed method discussed above (Claim 9), wherein all claim limitations have been

addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are

also obvious.

Per claim 31 (Currently Amended), this is the machine readable <u>storage</u> medium version

of the claimed method discussed above (Claim 10), wherein all claim limitations have been

addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are

also obvious.

Per claim 43 (New), this is the system version of the claimed method discussed above

(Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set

forth above. Thus, accordingly, these claims are also obvious.

Per claim 44 (New), this is the system version of the claimed method discussed above

(Claim 2), wherein all claim limitations have been addressed and/or covered in cited areas as set

forth above. Thus, accordingly, these claims are also obvious.

Per claim 45 (New), this is the system version of the claimed method discussed above

(Claim 3), wherein all claim limitations have been addressed and/or covered in cited areas as set

forth above. Thus, accordingly, these claims are also obvious.

Per claim 47 (New), this is the system version of the claimed method discussed above

(Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set

forth above. Thus, accordingly, these claims are also obvious.

7.   Claims 1, 3-10, 13-15, 22, 24, 26, 28, 30-31, 43-45 and 47 are rejected under 35 U.S.C.

103(a) as being unpatentable over Boykin et al. (US 2004/0123279 A1) in view of Berry et al.

(US 6,742,178 B1).

Per claim 1 (Currently Amended), Boykin discloses a <u>method for modifying a software</u>

<u>application</u>, comprising:

<u>modifying</u> a classfile after said classfile has been compiled from source code, <u>version of</u>

<u>software application</u> said classfile describing properties of a class within an object oriented

environment (e.g. FIG. 4B, 412, 414 and related text);

modifying a method information structure <u>for each method associated with the software</u>

<u>application</u> by adding byte code instructions to said method information <u>structure</u> (e.g. FIG. 11A-

11B and paragraph [0007] "… injector inserts hooks in the loaded class…" and paragraph [0055]

– [0055]), to cause a plug-in handler method <u>associated with a plug-in handler</u> to execute an

output function for <u>each</u> method<u>, the plug-in handler to record method information associated</u>

<u>with methods </u>(e.g. Table 9, line 12, receiving said identifier "handle"; line 20, invoke

'hooks.pos(handle, new Integer(returnValue))'; Table 6, details of "hooks.post", "probe.post",

which calculates/reports transfer rate about the "read" method); and,

~~adding a method information structure that includes byte code instructions for registering~~

~~the identities of said class and said method with a dispatch unit that is responsible for dispatching~~

~~an invocation to said plug-in module during runtime execution of said modified byte code, said~~

~~invocation directed to said dispatch unit from said added byte code instructions~~

compiling results of the modifying of the class file, the results including method

information, the method information including dependency hierarchical tree indicting

dependency order of the methods, and a time hierarchical tree indicting chronological order of

the methods (paragraph [0079] "… add the special instrumentation code and then recompile the

modified class…" and e.g. FIG. 4B and related text); and

filtering the method information via a filtering module, according to user preferences and

the dependency and time hierarchical tree (paragraph [0054] "… probed classes can be specified

individually or through filters…" and "… methods can be specified individually or through

filters such as all 'public' methods or all methods with parameter list and e.g. FIG. 11A and

related text),

Boykin does not explicitly disclose inserting functions calls at entry points and exit points

of each method associated with the software application via a byte code modifier. Nevertheless,

as evidenced by the teaching of Bley, it is known to insert function calls at entry and exit points

of each method via a byte code modifier (see at least e.g. Fig. 3, steps 306-308). Thus, it is

respectfully submitted that it would have been obvious to one skilled in the art at the time the

invention was made to insert function calls at entry and exit points of each method to track path

flow along with metric information related to the amount of resource consumed as once

suggested by Bley (see at least col.2:55-65).

Per claim 3 (Currently Amended), Boykin discloses the method of claim 2 <u>further</u>

<u>comprising:</u>

adding a field information structure <u>to the method,</u> said field information structure

describing a field that is to store a numeric identifier of said class (paragraph [0071] "… inserts

code at …" and e.g. FIG. 7, step 708).

Per claim 4 (Currently Amended), Boykin discloses the method of claim 3 wherein said

numeric identifier is provided to said class by a method of which <u>a</u> dispatch unit is comprised

(see e.g. FIG. 4, registry e.g. by providing an identifier and Table 12).

Per claim 5 (Currently Amended), Boykin discloses the method of claim 1 wherein a

portion of said byte code instructions that are added to said method are for causing said plug-in

module's handler method to provide <u>an</u> output function treatment in response to an entry point of

said method being reached (e.g. Table 9, line 12, receiving said identifier "handle"; line 20,

invoke "hooks.pos(handle, new Integer(returnValue)).

Per claim 6 (Currently Amended), Boykin discloses the ~~classfile modification~~ method of

claim 5 wherein said output function treatment is a function selected from the group consisting

of:

1) recording a time of entry for said method (Table 6 getEntries(); for (int i = 0)... i++);

2) recording an input parameter value for said method (Table 6 getEntries(); for (int i =

0)... i++); and,

3) incrementing a counter for said method (Table 6 getEntries(); for (int i = 0)... i++).

Per claim 7 (Currently Amended), Boykin discloses the classfile modification method of

claim 1 wherein a portion of said byte code instructions that are added to said method are for

causing said plug-in ~~module's~~ handler method to provide said output function treatment in

response to an exit point of said method being inevitably reached (e.g. Table 9, line 12, receiving

said identifier "handle"; line 20, invoke "hooks.pos (handle, new Integer(returnValue)).

Per claim 8 (Currently Amended), Boykin discloses the method of claim 7 wherein said

output function treatment is a function selected from the group consisting of:

1) recording a time of entry for said method (Table 6 getEntries(); for (int i = 0)... i++);

2) recording an input parameter value for said method (Table 6 getEntries(); for (int i =

0)... i++); and,

3) incrementing a counter for said method (Table 6 getEntries(); for (int i = 0)... i++).

Per claim 9 (Currently Amended), Boykin discloses the method of claim 7 wherein

portions of said byte code instructions that are added to said method are for causing said plug-in

module's handler method to provide said output function treatment in response to any exit point

of said method being inevitably reached (e.g. Table 9, line 12, receiving said identifier "handle";

line 20, invoke "hooks.pos(handle, new Integer(returnValue)).

Per claim 10 (Currently Amended), Boykin discloses the method of claim 1 wherein a portion of said byte code instructions that are added to said method are for causing said plug-in module's handler method to provide said output function treatment in response to an error arising during execution of said method (e.g. Table 9, line 12, receiving said identifier "handle"; line 20, invoke "hooks.pos(handle, new Integer(returnValue)).

Per claim 13 (Currently Amended), Boykin discloses the method of claim 12 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke special instruction (paragraph [0079] "… add the special instrumentation code and then recompile the modified class…" and e.g. FIG. 4B and related text).

Per claim 14 (Currently Amended), Boykin discloses the method of claim 1 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke virtual instruction (see Table 3 and paragraph [0079] "… add the special instrumentation code and then recompile the modified class…" and e.g. FIG. 4B and related text).

Per claim 15 (Currently Amended), Boykin discloses the classfile modification method of claim 1 wherein said byte code instructions are Java compatible and wherein said at least one of said instructions is an invoke special instruction (see Table 3 and paragraph [0079] "… add the special instrumentation code and then recompile the modified class…" and e.g. FIG. 4B and related text).

Per claim 22 (Currently Amended), this is the machine readable storage medium version of the claimed method discussed above (Claim 1), wherein all claim limitations have been

addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are

also obvious.

Per claim 24 (Currently Amended), this is the machine readable medium version of the

claimed method discussed above (Claim 3), wherein all claim limitations have been addressed

and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also

obvious.

Per claim 26 (Currently Amended), this is the machine readable storage medium version

of the claimed method discussed above (Claim 5), wherein all claim limitations have been

addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are

also obvious.

Per claim 28 (Currently Amended), this is the machine readable storage medium version

of the claimed method discussed above (Claim 7), wherein all claim limitations have been

addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are

also obvious.

Per claim 29 (Currently Amended), this is the machine readable storage medium version

of the claimed method discussed above (Claim 8), wherein all claim limitations have been

addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are

also obvious.

Per claim 30 (Currently Amended), this is the machine readable <u>storage</u> medium version of the claimed method discussed above (Claim 9), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 31 (Currently Amended), this is the machine readable <u>storage</u> medium version of the claimed method discussed above (Claim 10), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 43 (New), this is the system version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 44 (New), this is the system version of the claimed method discussed above (Claim 2), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 45 (New), this is the system version of the claimed method discussed above (Claim 3), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 46 (New), this is the system version of the claimed method discussed above

(Claim 4), wherein all claim limitations have been addressed and/or covered in cited areas as set

forth above. Thus, accordingly, these claims are also obvious.

Per claim 47 (New), this is the system version of the claimed method discussed above

(Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set

forth above. Thus, accordingly, these claims are also obvious.

Conclusion

8.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to ISAAC T. TECKLU whose telephone number is (571) 272-7957.

The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would

like assistance from a USPTO Customer Service Representative or access to the automated

information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/   /Tuan Q. Dam/
Examiner, Art Unit 2192   Supervisory Patent Examiner, Art Unit 2192